WHAT IS CLAIMED IS:

1. A method for determining the integrity of an application program running on a computer system having a memory, said application program having at least a data portion residing in the memory, the method comprising the steps of:

(a) pre-allocating one or more segments in said data portion;

(b) inserting tables in said segments;

(c) executing said application program on said computer system using an operating system, said application program produced by:

(c1) linking one or more relocatable object modules with one or more libraries and other object modules to form an intermediate executable module, said relocatable object modules being pre-compiled, and said libraries and said other object modules comprising relocation data,

(c2) examining said relocation data to determine selected addresses, said selected addresses corresponding to address locations in said segments,

(c3) storing said selected addresses in said tables,

(c4) storing a default address of a selected subprogram in said data portion, and

(c5) loading said libraries and said other object modules in said memory to transform said intermediate executable module into said application program executable by said computer system;

(d) determining a reference address associated with said selected subprogram at run-time for said application program;

(e) comparing said reference address with said default address; and

(f) executing a security application or module to determine said integrity of said application program based on said reference address and said selected addresses in said tables.

2. The method of Claim 1, wherein said step (f) uses said reference address if said reference address is equal to said default address.

3. The method of Claim 1, further comprising the step of computing a substitute address by offsetting memory locations of said selected addresses stored in said tables for every selected address.

4. The method of Claim 3, wherein said step (f) uses said substitute address if said reference address is unequal to said default address.

5. The method of Claim 3, wherein said selected addresses are offset by adding or subtracting an offset to said selected addresses.

6. The method of Claim 1, wherein said selected addresses are selected from a group consisting of memory references and jump target addresses, and said subprograms are selected from a group consisting of functions, subroutines, procedures and libraries.

7. The method of Claim 1, wherein said security application is a checksum application.

8. The method of Claim 1, wherein said security application decrypts previously encrypted data.

9. The method of Claim 8, wherein said data is encrypted while said tables are being inserted.

10. The method of Claim 1, wherein said application program comprises encrypted data residing on a DVD disk.

11. A computer system, comprising:

a central processing unit;

memory accessible by the central processing unit;

at least one application program executable on said central processing unit and within said memory; and

means for determining the integrity of said application program according to a method as described in Claim 1.

12. The computer system of Claim 11, wherein said step of executing a security application uses said reference address if said reference address is equal to said default address.

13. The computer system of Claim 11, further comprising the step of computing a substitute address by offsetting memory locations of said selected addresses stored in said tables for every said selected address.

14. The computer system of Claim 11, further comprising the step of storing said selected addresses in a compressed and encrypted format in said tables.

15. The computer system of Claim 13, wherein said step of executing a security application is based on using said substitute address if said reference address is unequal to said default address.

16. The computer system of Claim 11, wherein said application program comprises encrypted data residing on a DVD disk.

17. A computer readable medium having computer-executable instructions, which when executed on a computer system, causes said computer system to determine the integrity of an application program running on said computer system, said application program having at least a data portion residing in memory, said computer-executable instructions causing said computer system to perform the steps of:

(a) pre-allocating one or more segments in said data portion;

(b) inserting tables in said segments;

(c) executing said application program on said computer system using an operating system, said application program produced by:

(c1) linking one or more relocatable object modules with one or more libraries and other object modules to form an intermediate executable module, said relocatable object modules being pre-compiled, and said libraries and said other object modules comprising relocation data,

(c2) examining said relocation data to determine selected addresses, said selected addresses corresponding to address locations in said segments,

(c3) storing said selected addresses in said tables,

(c4) storing a default address of a selected subprogram in said data portion, and

(c5) loading said libraries and said other object modules to transform said intermediate executable module into said application program executable by said computer system;

(d) determining a reference address associated with said selected subprogram at run-time of said application program;

(e) comparing said reference address with said default address; and

(f) executing a security application or module to determine said integrity of said application program based on said reference address and said selected addresses in said tables.

18. The computer readable medium of Claim 17, further comprising the step of computing a substitute address by offsetting memory locations of said selected addresses stored in said tables for every said selected address, said step of executing a security application being based on using said substitute address if said reference address is unequal to said default address.

19. The computer readable medium of Claim 17, wherein said step (f) is based on using said reference address if said reference address is equal to said default address.

20. A method for determining the integrity of a relocatable application program executable on a computer system, said program being generated from one or more pre-compiled object files, said computer system including memory and said application

program having at least a data space residing in said memory, said method comprising the steps of:

(a) inserting tables into pre-allocated memory segments residing in said data space;

(b) examining relocation data for selected addresses when said pre-compiled object files are linked and loaded with one or more libraries and other object files, said libraries and said other object files comprising said relocation data;

(c) storing said selected addresses in said tables;

(d) storing a default address in said data space, said default address being associated with a point of reference within said pre-complied object files;

(e) determining a reference address from said application program at run time, said reference address corresponding to said point of reference;

(f) comparing said reference address with said default address; and

(g) performing a checksum to determine said integrity of said application program based on said reference address and said selected addresses in said tables.

21. The method of Claim 20, wherein said step of performing a checksum is based on using said reference address if said reference address is equal to said default address.

22. The method of Claim 20, wherein said step of performing a checksum is based on using a substitute address if said reference address is unequal to said default address, said substitute address being computed by offsetting memory locations of selected addresses for every selected address.

23. The method of Claim 22, wherein said offsetting is done by subtraction or addition.

24. The method of Claim 20, wherein said application program comprises encrypted data residing on a DVD disk.

25. A computer readable medium having computer-executable instructions, which when executed on a computer system may encrypt and/or decrypt a portion of an application program enabled to run on said system, and causes said computer system to determine the integrity of said application program having at least a data portion residing in memory, said computer-executable instructions causing said computer system to perform the steps of:

(a) pre-allocating one or more segments in said data portion;

(b) inserting tables in said segments;

(c) executing said application program on said computer system using an operating system, said application program produced by:

(c1) linking one or more relocatable object modules with one or more libraries and other object modules to form an intermediate executable module, said relocatable object modules being pre-compiled, and said libraries and said other object modules comprising relocation data,

(c2) examining said relocation data to determine selected addresses, said selected addresses corresponding to address locations in said segments,

(c3) storing said selected addresses in said tables,

(c4) storing a default address of a selected subprogram in said data portion, and

(c5) loading said libraries and said other object modules to transform said intermediate executable module into said application program executable by said computer system;

(d) modifying one or more portions of said segments by encryption;

(e) determining a reference address associated with said selected subprogram at run-time of said application program;

(f) comparing said reference address with said default address; and

(g) executing a security application or module to determine said integrity of said application program based on said reference address and said selected addresses in said tables.

26. The computer medium of Claim 25, wherein said method prevents access to encryption keys associated with said application program.

27. The computer medium of Claim 25, wherein said step of modifying one or more portions of said segments by encryption comprises the step of (a) adding or subtracting memory locations of said selected addresses in the encrypted object; (b) performing decryption; and (c) adding or subtracting memory locations of said selected addresses after decryption.